Lean Meets Agile:
Hybrid Project Management Methods

Dan Mason

Ceptara Corporation

CEPTARA

About Ceptara

- Ceptara Corporation
  - Founded 2002
  - A group of Black Belt LSS practitioners with broad industry experience, including government
  - Specializing in Lean Six Sigma Training, Project Management, and Six Sigma Projects and Initiatives
  - Certified for the Washington State “Lean Pool” of qualified vendors for State contracts
  - Personal Productivity add-ins for MS Outlook
  - http://www.ceptara.com/ 888.942.4625 (888 9 4A-GOAL)

CEPTARA

About Dan Mason

- 20+ years in Product Development – software and systems
- CSM and LSSBB certified
- Currently consulting on a Performance Improvement engagement
- Methodology geek – fascinated by the puzzle of high-performance software teams
- Explore every methodology option that pops up, as well as organizational development approaches

CEPTARA

About "Lean Meets Agile": Previous Agile Friday Webinars

- A wealth of information on Lean-Agile subjects has already been delivered
- Excellent presenters with wonderful experience have related their learnings
- Agile Fridays website has recordings of most sessions – check it out
- "Hybrid Lean-Agile" is obviously not a unique topic – it seems to be THE topic

CEPTARA

My Approach Today

- Historical view of Lean-Agile emergence
- Contrast Lean-Agile with Waterfall
- Note the commonalities between Lean and Agile
- First Principles – the fundamentals of why Lean and Agile actually work
- Why you may want to try Agile/Scrum first – and why not
- Show how you can mix certain practices from Lean-Kanban and Agile-Scrum (hybrid approaches)
- Leave you with reference materials – books, essays - for further exploration

Quick note on terminology

- Scrum and Agile are often used interchangeably
  - Scrum is a specific Agile project management method
  - There are other Agile methods (e.g. Extreme Programming)
  - Scrum is Agile but not all Agile is Scrum
  - The “Agile Manifesto” is specifically about software
- Lean and Kanban are often used interchangeably
  - Software Kanban is a specific “Lean Software” technique
    - Kanban is also a mechanism for controlling manufacturing flow
  - “Lean” is a much broader umbrella term
  - “Lean Thinking” is a way of approaching systems analysis
- For today, we’ll use Scrum as the proxy for Agile, and Kanban as the proxy for Lean Software

CEPTARA

Why such interest in Lean-Agile?

- Spotty success with variants of Waterfall method for a long time – a very long time.
- The management of software development has continues to undergo profound changes in search of silver bullet
- Why can’t we figure it out?
  - Fundamental difficulties in managing software have been recognized and battled for decades, without resolution
  - See writings of Fred Brooks, Tom DeMarco, and others
  - 40 years of searching for the One Way
  - No “one size fits all” technique has yet been devised, though there are passionate proponents of Agile and Lean

CEPTARA

Fred Brooks's Insight on Planning

"Even the best planning is not so omniscient as to get it right the first time"
Fred Brooks, The Mythical Man-Month, 1975 (yes, 40 years ago)

- Mr. Brooks was right. Yet, for many years, we've been trying to predict what will happen during a software project, as if we had perfect knowledge
- With the Predictive approach, we try to predict what we will build, all the activities involved in building it, and how long these activities will take
- This hasn't worked so well. Only 20-30% of projects come in on plan
- We have learned that even if we had perfect knowledge TODAY for planning, the same knowledge will not be perfect tomorrow
- Circumstances change in multiple dimensions, from requirements definition to knowledge level of programmers

CEPTARA

Fred Brooks, continued…

"The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures. (This very tractability has its own problems.)"

- Sounds a little grandiose, but the point is that *software goes almost directly from someone's head to a working system. Large changes can be made with a few keystrokes.* How wonderfully flexible, yet incredibly unpredictable can you get?
- Furthermore, each day what's in the programmer's head changes, due to gaining knowledge and skill, both generally and in the specific problem domain…every day it's a new ball game
- A team of programmers is itself a *complex adaptive system* – it learns from experience and adapts to this new understanding as it goes
- The more often you can recalibrate the plan based on new knowledge, the more likely you are to constantly be on the optimal path

CEPTARA

"First Principles" of Lean and Agile

- Lean and Agile are both examples of "empirical process control"…
  - Both of approaches seek constant, quick incorporation of new knowledge to inform the next steps – your process is guided by the reality you encounter along the way
  - Both employ the "Deming Cycle" PDSA (Plan-Do-Study-Act) to constantly recalibrate the plan based on observed results
  - As opposed to "predictive process control", where you make a prediction, then strive to "bend reality" to make the prediction true by juggling resources, redefining success and generally driving to the plan rather than the desired outcome

First Principles of Lean and Agile

- To incorporate reality, Lean and Agile both emphasize the delivery of small increments of software, frequently…
  - Lean (Kanban) does this using one-piece flow and reduction of "cycle time" by eliminating wasteful activities
  - Agile (Scrum) does this by focusing on delivery of new working code, incrementally improved, defined for every sprint (iteration)
  - Both approaches ensure that new learning may be constantly applied
- Limit the amount of work-in-progress
  - Scrum does this using Sprints, timeboxes, and backlogs
  - Kanban does this by setting explicit WIP limits on process steps
  - Small batches work better than large
- High degree of collaboration/communication
  - Scrum has self-organizing teams and daily stand-up meetings
  - Kanban has no prescribed roles, and a *kaizen* approach
  - Both methods rely on face-to-face communication rather than document handoffs

CEPTARA

First Principles (continued)

- PULL work into a flow, don’t push it…
  - Like a wet noodle…pushing is not effective
  - When work is pulled though a system, it flows smoothly
  - When we attempt to push it, it simply backs up – and queues of work are very bad
  - Agile/Scrum pulls from a prioritized “backlog” of work (stories) for each sprint
  - Lean/Kanban pulls from a queue of work on demand, whenever capacity is available (no sprints)

CEPTARA

So which is better, Lean or Agile?

- Both. (are better than predictive methods…)
- They are more similar than different, especially relative to typical waterfall sequential methods featuring stage-gate flow and big-up-front-planning
- Why get religious about it? Use what works for your situation! One from Column A, one from Column B
- Some teams will thrive with Scrum, others will reject it in favor of a less prescriptive Lean approach
- What “works” is dependent on culture, team personalities, maturity of existing capability, type of work product, size of projects – in other words CONTEXT

Scrum can be good starting point…

- There is value in starting with "pure" Scrum – especially if you need to break old ways
- Scrum is ***much*** more rigid about roles, rituals, and artifacts than Lean/Kanban. This can be useful, because Scrum's rules introduce useful habits:
  - High degree of collaboration and communication encouraged via prescribed meetings (like daily stand-ups)
  - Total focus on delivering a limited set of product features in a short timeframe, discouraging distractions
  - "Visual controls" including highly visible metrics (burndown charts) and "scrum boards" for day-to-day operations
  - Adjustment of plan frequently – every sprint has a new plan

CEPTARA

…But Scrum has downsides…

- Very dramatic cultural change for staff (tightly prescribed roles and rituals)
- Tends to optimize the dev team and ignore other parts of the company – doesn't always work in the real world
- The structure of Scrum does not scale well to larger teams and projects – sweet spot is 5-7 people
- Sprint structure can perpetuate faults of Waterfall – deadlines can lead to same old problems with estimation sandbagging, cutting corners to meet dates, buildup of technical debt
- Can become a set of "little waterfalls"

CEPTARA

Lean/Kanban starts more gradually

- Start with your current process and make incremental changes
- Understand your Value Stream by mapping your process and eliminating unnecessary work and “waste”
- Make all work visible and work states transparent
- Introduce “WIP Limits” (Work In Process) in your work states, i.e. how many items can be actively worked in each process step
- Measure “cycle time” to finish a feature or other unit of functionality, then improve upon it

CEPTARA

Examples of “Hybridizing” Methods

- Changing the daily stand-up meeting (Scrums)
  - Daily meetings are good discipline but sometimes are too much
  - Some teams change “cadence” to every other day, twice a week
- Time-boxing, or not
  - Short time-boxed sprints are encouraged by Scrum
  - The shortest possible sprint is one item at a time, which is essentially same as Kanban
  - Some teams use a Kanban approach without time-boxes, but have planning and retrospective meetings on some regular schedule
- Alternatives to Burn-Down chart
  - Cumulative Flow Diagram can be used for Scrum
  - Prioritizing queues vs. FIFO

CEPTARA

More “Hybridizing” Methods

- Scrum generally tries to break down work into like-sized tasks so that “velocity” can be calculated more easily and items can be fitted into a sprint time-box.
  - Velocity is not absolutely necessary if you can estimate cycle time

- Scrum asks team to “commit” to a certain amount of deliverables in sprint
  - Commitment is a good thing, however it can lead to fudging to meet goals
  - Some flexibility in sprint deliverables can help
  - In the end, what matters is the team operating at max sustainable performance

Summary

- Lean/Kanban and Agile/Scrum are manifestations of similar principles
- Both methods are rooted in empiricism, rather than command-and-control
- As long as First Principles are met, a mix of techniques can be borrowed from each method
- Experiment, but drop things that don't work for your context

Fundamental Texts in Agile and Lean

- "Agile Project Management with Scrum"
  - Ken Schwaber
- Various Agile books by Mike Cohn, Kent Beck, Martin Fowler, Jim Highsmith
  - Can't go wrong with any of these
- "Kanban"
  - David Anderson
- "Lean Project Management, an Agile Toolkit"
  - A series of Lean PM books
  - Mary and Tom Poppendieck

CEPTARA

On Combining Scrum and Kanban..

- “Scrum and Kanban” by Henrik Kniberg
  - Easily understood, real-world examples, and CARTOONS
  - Practical hands-on examples and case study
  - Excellent distillation of Scrum-Kanban principles to their essence
- “Scrumban” by Corey Ladas
  - Essay that defined a new methodology
- “Principles of Product Development Flow” by Donald Reinertsen
  - Shows the math (queuing theory) behind many Lean principles such as small batch size, short iterations
  - Explains concepts such Cost of Delay and Weighted Shortest Job First

CEPTARA

Questions?

CEPTARA